**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Análisis y desarrollo de *software* |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501096. Desarrollar la solución de *software* de acuerdo con el diseño y metodologías de desarrollo. | RESULTADOS DE APRENDIZAJE | 20501096-04. Codificar el *software* de acuerdo con el diseño establecido. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 21 |
| NOMBRE DEL COMPONENTE FORMATIVO | Construcción API |
| BREVE DESCRIPCIÓN | En este componente formativo se abordarán los conceptos relacionados con la configuración del entorno de desarrollo como espacio de trabajo para la API REST. Se revisarán las buenas prácticas de los estándares de codificación para el desarrollo de la API, la cual será construida con las tecnologías de Node.js, Express y MongoDB y se abordará el *renting* de la API creada, a través de la herramienta Postman. |
| PALABRAS CLAVE | API, buenas prácticas, desarrollo, tecnologías |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**

# **La internet y la web**

## 1.1 Conceptos sobre el entorno de desarrollo integrado - IDE

## 1.2 Instalación del Visual Studio Code

## 1.3 Instalación y configuración del Node.js y npm

1.4 Instalación de módulo

# **Codificación de la API REST con Node.js**

## 2.1 Buenas prácticas en la codificación de Node.js

# ***Testing* de la API REST con Postman**

1. **INTRODUCCIÓN**

En este componente formativo se estudiarán los conceptos y pasos para el desarrollo de una API REST, utilizando las tecnologías del *back-end* como Node.js, Express y el sistema de bases de datos no relacional como lo es MongoDB. De esta forma se inicia con la puesta a punto del entorno de desarrollo con Visual Studio Code, el cual será el espacio de trabajo donde se construirá la API. Adicionalmente, se comprenderá la importancia del uso de los comandos (en la terminal) para la instalación y configuración de paquetes necesarios para el correcto funcionamiento del proyecto; de igual manera, se ilustran algunas buenas prácticas de desarrollo y herramientas que harán más prolijo el código.

Una vez construida la API se inicia con un *testing* a través de la herramienta Postman, que es muy utilizada para este tipo de desarrollos. Así, se corroboran las respuestas y solicitudes de un tipo de proyecto de esta naturaleza.

1. **DESARROLLO DE CONTENIDOS:**

# **La internet y la web**



Para todo proyecto *software* es recomendable alistar las herramientas necesarias y mínimas para iniciar con un desarrollo. En ese sentido, dependiendo del alcance, la tecnología y/o lenguaje se sugiere definir muy bien este tipo de recursos para realizar el ejercicio. En este orden de ideas se abordarán algunos conceptos fundamentales.

## **1.1 Conceptos sobre el entorno de desarrollo integrado - IDE**

Como primera medida, es necesario tener en cuenta la definición de un entorno de desarrollo integrado IDE, que es un espacio de trabajo basado en tecnologías y *software*, que sirven de plataforma para el diseño de aplicaciones que componen una serie de herramientas para el desarrollador de *software*. Un IDE cuenta con las siguientes características:

|  |
| --- |
| CF21\_1\_1\_Caracteristicas |

Para el caso de este componente formativo se utilizará el IDE Visual Studio Code por las siguientes razones:

* Visual Studio Code resalta las palabras clave del código en diferentes colores para ayudar a identificar fácilmente los patrones de codificación y aprender más rápido.
* A medida que se codifica, Visual Studio Code brinda sugerencias para completar líneas de código y soluciones rápidas para errores comunes.
* Se aprovecha porque este entorno tiene incorporado la terminal, lo cual hace mucho más fácil el desarrollo sobre todo con las tecnologías de Node.js, porque trae los paquetes necesarios para el proyecto.

En el proceso de alistamiento del entorno para iniciar el proyecto de desarrollo de la API, se deberá contar con los siguientes recursos:

|  |
| --- |
| CF21\_1\_1\_Recursos |

## 

## **1.2 Instalación del Visual Studio Code**

Para la instalación se deben seguir los siguientes pasos (para este caso se ha tomado como plataforma el sistema operativo Windows 10):

|  |
| --- |
| CF21\_1\_2\_Instalacion\_VSC |

## **1.3 Instalación y configuración del Node.js y npm**

Node.js está considerado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, de igual manera, está diseñado para crear aplicaciones *network* escalables y pensado en principio como tecnología *back-end* (node.js.org, 2021).

Los pasos para instalar el Node.js son los siguientes:

|  |
| --- |
| CF21\_1\_2\_Instalacion\_Node |

**1.4 Instalación de módulo**

Un paquete en Node.js contiene todos los archivos que se necesitan para instalar un módulo; los módulos son bibliotecas de JavaScript que se pueden incluir en el proyecto. Instalar módulos es una de las principales tareas que debe aprender a hacer al comenzar con el administrador de paquetes Node. El comando para instalar un módulo en el directorio actual es el siguiente:

$ npm install < módulo >

$ npm i < módulo >

El primer módulo a instalar es Express, el cual es el *framework* web más popular de Node y proporciona mecanismos para (Mozilla, 2021):

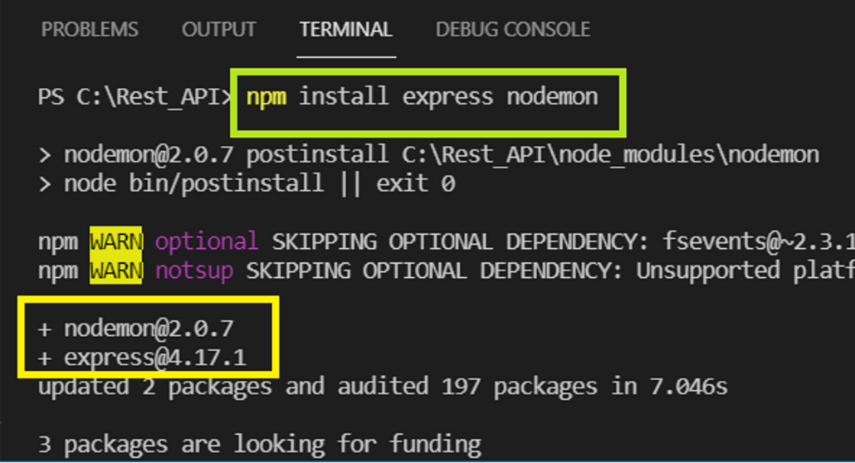
1. Escritura de manejadores de peticiones con diferentes verbos HTTP, en diferentes caminos URL (rutas).
2. Integración con motores de renderización de “vistas”, para generar respuestas mediante la introducción de datos en plantillas.
3. Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
4. Añadir el procesamiento de peticiones “middleware” adicional en cualquier punto dentro de la tubería de manejo de la petición.

El segundo módulo para instalar es nodemon, que es una utilidad que monitorea los cambios en el código fuente que se está desarrollando y de manera automática reinicia el servidor. Es una herramienta muy útil para el desarrollo de aplicaciones en Node.js (Gómez, 2017).

A continuación, se muestra la línea de comando que se debe ejecutar desde la terminal del Visual Studio Code.

**Figura 1**

Comando para la instalación de Express y nodemon



Hasta esta parte se ha abordado la instalación y configuración del espacio de trabajo para iniciar con el desarrollo de la API. No obstante, es necesario revisar las buenas prácticas de codificación para que el desarrollo no solo sea prolijo en su estructura, sino que también siga estándares de codificación, permitiendo que el código tenga un alto desempeño.

# 

# **Codificación de la API REST con Node.js**

En este tema, se inicia con el desarrollo de la API REST, por lo que se verá la codificación de cada uno de los archivos que componen el proyecto. Para esto, es importante tener en cuenta dos cosas, la primera es realizar buenas prácticas de codificación, permitiendo que el código quede legible y mucho más limpio al momento de documentar y la segunda, es entender la estructura de directorios del proyecto, la cual va a dar el orden al proceso de búsqueda e indexación de las rutas y recursos que se necesitarán al momento del desarrollo.

## **2.1 Buenas prácticas en la codificación de Node.js**

El objetivo de seguir estándares de codificación, es alcanzar un código mantenible y se considera mantenible cuando el código es legible, consistente, predictivo y documentado. En la práctica, la corrección de errores y modificación del código desarrollado de un producto *software* toma demasiado tiempo y se llega algunas veces a que no es posible realizar del todo el desarrollo, por lo que se toma la decisión de iniciar de cero, debido a las malas prácticas. (Choque, 2016)

A continuación, se indican las buenas prácticas enunciadas por uno de los mejores servicios de control de versiones y desarrollo de *software* colaborativo basado en Git, llamada GitLab. Esta comunidad sigue de igual manera estándares como Google JavaScript Style Guide (Whyte, 2021).

|  |
| --- |
| CF21\_2\_1\_Buenas\_practicas |

## **Desarrollo de la API REST**

Lo invitamos a revisar el PDF **Desarrollo API REST**, en el cual se describen de manera sucinta, los pasos requeridos para el desarrollo de la API REST con Node.js.

# ***Testing* de la API REST con Postman**

Con el fin de revisar y evaluar el funcionamiento de la API, se utilizará la herramienta Postman, que es una herramienta que se utiliza especialmente para el *testing* de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de tecnologías.

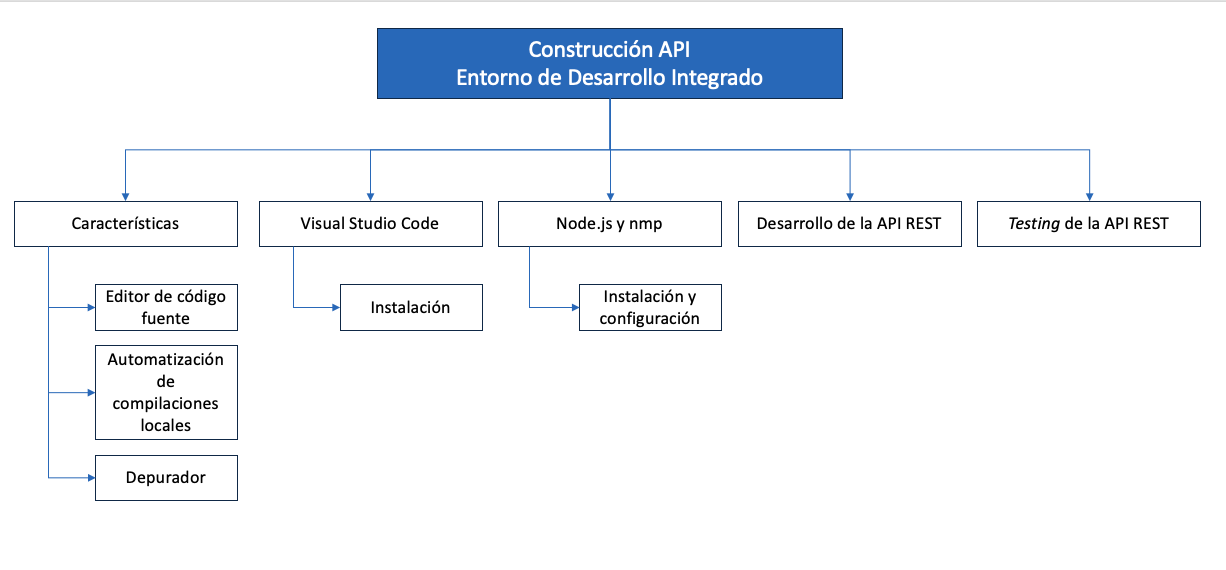
Con Postman, además de testear, consumir y depurar API REST, se puede monitorearlas y escribir pruebas automatizadas para ellas, documentarlas y simularlas. Esta herramienta es favorable para equipos con poca experiencia en programación y para hacer testing de todo tipo en general de API REST (López, 2021).

**Instalación Postman**

Lo invitamos a revisar el PDF **Instalación herramienta Postman**, en el cual se explica su instalación.

1. **SÍNTESIS**

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo:



1. **ACTIVIDADES DIDÁCTICAS (Se debe incorporar mínimo 1, máximo 2)**

Por cada componente formativo se puede proponer **un máximo de dos actividades** (mínimo debe ser 1) que los aprendices puedan realizar una vez han revisado los contenidos presentados y que refuercen la asimilación de los mismos. Entre las actividades propuestas se deben abordar todos los temas principales.

**Son actividades que no generan evaluación y que funcionarán independiente del LMS.**

En este ítem deberá diligenciar la siguiente tabla, que especifica las plantillas de diseño de actividad de afianzamiento que entregará para su incorporación en el ambiente virtual.

Para cada actividad debe indicar:

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Conceptos importantes |
| Objetivo de la actividad | Afianzar algunos de los conceptos más importantes asociados a la construcción de una API REST. |
| Tipo de actividad sugerida | Arrastar y soltar |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | CF21\_Actividad\_didactica.docx |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Entidades de información digital | Coding, G. (2020). Cómo instalar Node.js en Windows 10 [video]. YouTube. | Video | <https://www.youtube.com/watch?v=BgtB31gXkoA> |
| Información digital | Dominicode. (2018). Cómo usar Postman, en español [video]. YouTube. | Video | <https://www.youtube.com/watch?v=qsejysrhJiU> |
| Entidades de información digital. | DiSimone, K. (2020). mLab to Atlas for existing NS users [video]. YouTube. | Video | <https://www.youtube.com/watch?v=hXSLBACjYQo> |

1. **GLOSARIO:**

Incorpore aquí las definiciones de los términos claves, requeridas para comprender adecuadamente los contenidos de este recurso educativo. Presentarlo en Orden Alfabético. *Máximo 15 palabras*.

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Identación: | dentro del contexto de desarrollo de *software* identar es un término que significa mover un bloque de código hacia la derecha insertando espacios o tabulaciones, para así separarlo del margen izquierdo y distinguirlo mejor del texto adyacente, es decir, organizar mejor el código. La identación se considera un principio fundamental de buena práctica de la programación de software. |
| JSON: | *Java Script Object Notation* (JSON) es un formato basado en texto estándar que se utiliza para representar datos estructurados en la sintaxis de objetos de JavaScript. Se utiliza principalmente para transmitir datos en aplicaciones web (como es el caso de enviar algunos datos desde el servidor al cliente, por lo tanto, estos datos pueden ser mostrados en páginas web o viceversa). |
| *Localhost:* | en el contexto de las redes de computadoras, *localhost* es una computadora que ejecuta un programa. La computadora que se tiene al frente funciona también como un servidor virtual; este modo es comúnmente utilizado para realizar pruebas en el desarrollo web. |
| Manejadores o *handlers* de eventos: | en la programación con JavaScript un evento por sí mismo carece de utilidad; para que los eventos resulten útiles o aplicables se debe asociar funciones o código JavaScript a cada evento. De esta forma, cuando se produce un evento de manera inmediata se ejecuta el código indicado, por lo que la aplicación responde ante cualquier evento que se produzca durante su ejecución. Las funciones o código JavaScript que se realizan para cada evento se denominan "manejador de eventos”. |
| *Middleware:* | es un bloque de código que se ejecuta entre la petición que hace el usuario (*request*) hasta que esa petición llega al servidor. De esta manera son funciones que tienen acceso al objeto de solicitud (req), el objeto de respuesta (res) y la siguiente función de *middleware* en el ciclo de solicitud-respuesta de la aplicación. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Choque, J. C. (2016). Adición de estándares de codificación node.js. GitLab <https://gitlab.agetic.gob.bo/dbarra/bolivialibre/commit/1ff84b6ae70f0e9436f6e6ddc7f3e060bc9556ad>

Chrome web store. (2021). Postman <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=es-419>

Express. (2021). Direccionamiento básico. <https://expressjs.com/es/starter/basic-routing.html>

Gómez, D. (2017). Nodemon. Vortexbird <https://vortexbird.com/nodemon/#:%7E:text=Nodemon%20es%20una%20utilidad%20que,desarrollo%20de%20aplicaciones%20en%20nodojs>

López, A. (2021). Qué es Postman y para qué sirve. OpenWebinars https://openwebinars.net/blog/que-es-postman/

Mozilla. (2021). Await. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/await>

Mozilla. (2021). Haciendo la programación asíncrona más fácil con async y await. MDN Web Docs. <https://developer.mozilla.org/es/docs/Learn/JavaScript/Asynchronous/Async_await>

Mozilla. (2021). Introducción a Express/Node. MDN Web Docs. <https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction>

Nodejs.org. (2021). Acerca de Node.js. Node.js https://nodejs.org/es/about/

npm. (2021). About npm. <https://www.npmjs.com/about>

Red Hat. (2021). El concepto de IDE. <https://www.redhat.com/es/topics/middleware/what-is-ide>

Whyte, A. (2021). Google JavaScript Style Guide. <https://google.github.io/styleguide/javascriptguide.xml>

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha |
| Autor (es) | Zulema Yidney León Escobar | Experta temática | Regional Cauca - Centro de Teleinformática y Producción Industrial |  |
| Jonathan Guerrero Astaiza | Experto temático | Regional Cauca - Centro de Teleinformática y Producción Industrial |  |
| Ana Catalina Córdoba Sus | Evaluadora instruccional | Regional Antioquia - Centro de Servicios de Salud | Marzo 2024 |
| Olga Constanza Bermúdez Jaimes | Responsable Línea de Producción Antioquia | Regional Antioquia - Centro de Servicios de Salud | Marzo 2024 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |

**Nota:**Para la propuesta instruccional se deben tener en cuenta las métricas desarrolladas en el equipo:

<https://drive.google.com/drive/u/1/folders/1UiJvaklSCICR4BaQ7ga_q04JFa53h_u_>